

Silent Data Corruption: A Threat to Data Integrity in High-End Computing Systems

Sarah Michalak

Statistical Sciences Group

Los Alamos National Laboratory

michalak@lanl.gov

Collaborators

- Sean Blanchard
- Carolyn Connor
- John Daly
- Andy DuBois
- Dave DuBois
- Andrea Manuzzato
- Dave Modl
- Lisa Moore
- Heather Quinn
- Bill Rust
- Andrew Shewmaker
- And many others...

LANL HPC Platforms and Users

- **Capability and capacity systems at LANL, Lawrence Livermore National Laboratory, and Sandia National Laboratory are available to LANL scientists**
- **Used for large scientific calculations:**
 - Capability: large jobs that run for several months
 - Capacity: smaller jobs that can run for up to several months
- **Some HPC platforms are at the cutting edge in terms of scale/technology**



Silent Data Corruption (SDC)

- **“SDC occurs when incorrect data is delivered by a computing system to the user without any error being logged” Cristian Constantinescu (AMD)**
- **Examples of SDC:**
 - Teraflops Supercomputer [Constantinescu, 2000]
 - Processor ECC disabled (firmware bug)
 - UWI Study [Kola et al (2005)]
 - Faults in storage and in staging/compute nodes lead to SDC
 - Some root causes developed over time, so not observed during initial testing
 - CERN FS Study [Panzer-Steindel (2007)]
 - Disk Errors: write, read, compare 2 GB file
 - Every 2 hrs for 5 weeks on 3000 nodes → 500 errors on 100 nodes
 - Recalculate and compare checksum for 33700 files (~8.7 TB)
 - 22 mismatches → one bad file in 1500
 - LANL Testing of Decommissioned Platform
 - 70 incorrect Linpack results; all involve a single node

What We Know or Think We Know

- **SDC can affect networks, nodes, and file systems**
- **SDC has multiple causes** [Constantinescu 2008]
 - Temperature/voltage fluctuations, particles, manufacturing residues, oxide breakdown, electro-static discharge...
- **SDC will likely be more prevalent in new technologies** [Borkar 2009; Pan et al 2008; Constantinescu 2006]
 - Increased frequency, transistor counts, soft errors, and noise levels
 - Decreased feature sizes and supply voltage
- **For a given device susceptibility, a larger platform is more likely to be affected**
- **SDC could affect scientific desktop computing**
 - Laptops/desktops used for scientific computation may be equivalent to a cluster
- **Applications have differing susceptibility to faults that could lead to SDC**
 - Not all undetected faults will lead to SDC

LANL Efforts

■ Platform Testing:

- Production Platforms:
 - Rolling out Linpack-based testing on all production platforms
 - OSATSDC software available
 - No SDC observed as yet
- Decommissioned Platforms:
 - Test at nominal or manipulate temperature/voltage
 - SDC observed on one platform

■ Laboratory Testing:

- Dual-core 65nm processor on a high-performance overclocking motherboard
- Manipulate frequency, voltage, fan speed/temperature while running Linpack
- Multiple forms of SDC: incorrect Linpack results, erroneous timestamps/environmental data
- Other Errors: program termination with no error, program crash, system crash

LANL Efforts (cont'd)

■ Neutron-Beam Testing:

- Test HPC-relevant hardware
- Run different applications and collect hard fail and SDC data
- Research questions:
 - Do the hard fail and SDC rates depend on the application?
 - Does manufacturing variability affect the hard fail and SDC rates?
 - Do frequency, voltage, and temperature affect the hard fail and SDC rates?
 - For later study

■ Future:

- E2E testing of HPC resources
- Development of test codes
 - Efficient and effective test codes for different subsystems/components
- Environmental testing (temperature, particles, ...)
- Investigation of application susceptibility

Operational Questions of Interest

Goal: Understand & mitigate the impact of SDC on users, e.g.:

- **What is the probability that my code that runs for h hours on n nodes on a particular platform gets the wrong answer?**
- **What is the probability that if I write my data to disk and then read it back again, I get the same results?**
- **What operational strategies can mitigate the impact of SDC on users?**

These operational considerations lead to research questions.



IM-4-RN01-006-045

System-Focused Research Questions & Required Information I

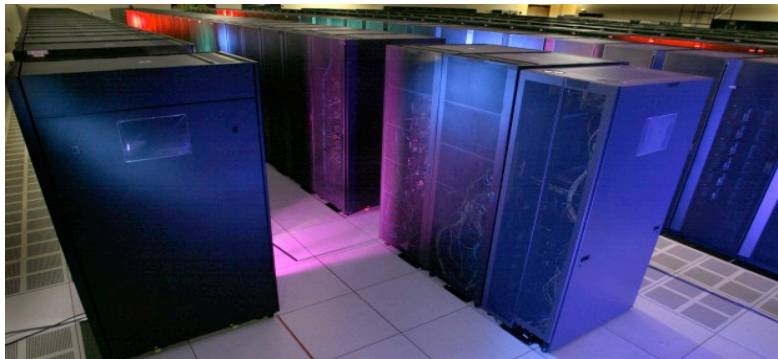
- **What are the causes of SDC? How do they change with technology? What can I know today about trends for future technologies?**
 - Enable system planning and resilience work
 - Challenge: Multiple diverse causes occurring in complex devices that evolve with time
- **What is the distribution of undetected-error rates for a particular architecture? Can I know this before a system is built?**
 - How bad could it get? Large systems have 10,000s of components, so some will be less robust. How can we find the “bad” components?
 - Challenge: Understanding manufacturing and other sources of variability that can affect susceptibility to SDC; need for testing and modeling to do so?
- **Can I determine $\text{Pr}(\text{SDC}) = f(\text{temp, volt, freq, particles, ...})$?**
 - Identify that a device has crossed a threshold (aging, temperature, voltage, ...) making SDC more likely
 - Enable mitigation strategies; describe behavior outside of nominal
 - Challenges: Methods for testing/modeling/extrapolation/UQ for possibly sparse data; understand, detect, and provide information about precipitating conditions

System-Focused Research Questions & Required Information II

- **What methods can be used to perform accelerated testing to learn about SDC rates under nominal conditions?**
 - Beam-testing may not be sufficient
 - Challenge: Development of such tests, including a sound method of extrapolating to nominal conditions for applications/workloads of interest
- **How can production systems be efficiently and effectively tested to identify problems *before* they affect users?**
 - Initial and on-going testing to protect user computations and data
 - Challenges: Development of E2E testing that is sensitive to faults that could cause SDC; balance between production use and testing
- **Can I know that a fault that could cause SDC has occurred?**
 - Focused use of fault-mitigation technologies
 - Challenge: Detect & report faults that could cause SDC

Application-Focused Research Questions & Required Information

- **How do I convert from an undetected-error rate to a SDC rate for a particular application?**
 - Understand the impact of undetected errors on applications of interest
 - Challenge: Quantify application-specific vulnerability
- **How can I make my application more resilient to SDC?**
 - Ensure computational correctness despite undetected errors
 - Challenge: Development of application-specific fault detection and resilience techniques
- **What can I do to help an arbitrary code?**
 - Challenge: Development of generic fault detection and resilience techniques



UNCLASSIFIED

Conclusion

- **Large-scale platforms will continue to be used for scientific computation**
 - Increased use of large-scale simulation for decision-making
- **Users need to get correct answers and the resulting data/output needs to have its integrity maintained**
- **Integrity of computations performed on desktop computers/laptops must also be ensured**
- **“Perfect” systems are likely unrealistic**
- **Thus, research is required to:**
 - Understand the causes of and characteristics of faults that could lead to SDC
 - Mitigate the impact of SDC on users
 - Research on resilient methods requires information about SDC rates
- **Any solutions need to be mindful of power, financial, and other constraints/requirements on the system**